Jacob Kirby

jak307@pitt.edu

INFSCI 1599: Intermediate Programming with Python

Dmitriy Babichenko

December 7, 2022

**Table of Contents**

Introduction/Abstract – Page 3

E-R Diagram – Page 4

Business rules – Page 5

UML Class Diagram – Page 7

Closing Section – Page 8

**Introduction/Abstract**

For my final project, I decided to make a database for a type of social networking service aimed at film enthusiasts. Using the application that I built to interact with the database, a user could create an account, search for films from the database and find the names of key players involved with its creation along with a plot summary and some other basic information about the film. They could then choose to write a review of the film, respond to other users’ reviews, or add the film to a list.

A user of the application could also search for other users’ profiles, view and respond to their activity, and follow them. Additionally, a user can view their own profile and delete/edit their own reviews, responses, and lists.

The purpose of this project, if it were to be more extensive, would be to help people find films that they may enjoy by following the activity of other like-minded users.

**E-R Model**

**Diagram

Description automatically generated**

**Business Rules**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity 1** | **Entity 2** | **Cardinality on Entity 1 side** | **Cardinality on Entity 2 side** | **Business Rule(s)** |
| Film | Director | 1..\* | 1..\* | A film may have one or many directors. A director may have one or many films. |
| Film | Writer | 1..\* | 1..\* | A film may have one or many writers. A writer may have one or many films. |
| Film | Producer | 1..\* | 1..\* | A film may have one or many producers. A producer may have one or many films. |
| Film | Actor | 1..\* | 1..\* | A film may have one or many actors. An actor may have one or many films. |
| Film | Cinematographer | 1..\* | 1..\* | A film may have one or many cinematographers. A cinematographer may have one or many films. |
| Film | Editor | 1..\* | 1..\* | A film may have one or many editors. An editor may have one or many films. |
| Film | Composer | 1..\* | 1..\* | A film may have one or many composers. A composer may have one or many films. |
| Film | Studio | 1..\* | 1..\* | A film may have one or many studios. A studio may have one or many films. |
| Film | Genre | 1..\* | 1..\* | A film may have one or many genres. A genre may have one or many films. |
| User | Review | 1..1 | 0..\* | A user may have zero or many reviews. A review must have one user. |
| User | List | 1..1 | 0..\* | A user may have zero or many lists. A list must have one user. |
| Film | Review | 1..1 | 0..\* | A film may have zero or many reviews. A review must have one film. |
| Film | List | 1..1 | 0..\* | A film may have zero or many lists. A list must have one film. |
| Review | Review reply | 1..1 | 0..\* | A review may have zero or many review replies. A review reply must have one review. |
| List | List reply | 1..1 | 0..\*S | A list may have zero or many list replies. A list reply must have one list |
| User | User | 0..\* | 0..\* | A user may be followed by zero or many users. A user may follow zero or many users. |

**Class Diagram**

**Diagram, schematic

Description automatically generated**

**Closing Section**

My experience with the project was arduous, yet relatively smooth. I didn’t run into any significant issues that required intense problem-solving, but the experience of working with a database did feel tedious. I think this has showed me that I’ll probably need a lot more practice working with databases in order to code more efficiently.